UDC 004.94

Vitalii Omelchenko, Oleksandr Rolik

FORECASTING-AT-SCALE ALGORITHMS FOR PREDICTION CLUSTER WORKLOAD

The paper deals with the issue of predicting workloads in a cluster to use in proactive scaling of computing resources. Forecasting-at-scale algorithms Prophet and Greykite for forecasting time series are considered, their accuracy and universality are evaluated.

Keywords: resource management, proactive scaling, Kubernetes, automatic scaling, Prophet, Greykite. Fig.: 4. Tab.: 1. Bibl.: 10.

Problem statement. When managing the quality of services provided by clouds or enterprise-level IT infrastructures, it is important to maintain the quality of services at an agreed level [1]. In general, maintaining the quality of services at the agreed level is ensured by providing additional computing, communication, and other resources to those applications that provide the corresponding service. Effective management of the quality of services can be ensured mostly by automatic adding or reducing the number of resources. For this purpose, data center computing resource management systems of cloud service providers or corporate IT infrastructures contain modules that use automatic scaling methods and tools, taking into account service delivery technologies. The autoscaling methods that use the results of load forecasting will be more effective from the point of view of maintaining the quality of services at the agreed level. Modern paradigms of service provision are built on the use of microservices, containers, etc. The popularity of microservices in the provision of services entailed the creation of various technologies for the implementation of microservice architecture, including those based on clusters. At the same time, it is advisable to solve the problems of maintaining the quality of services at the agreed level by scaling all components of the cluster. It is necessary to take into account the fact that each component of the microservice architecture has its unique features of work and functionality, as well as the fact that the load pattern is individual for each component, and there can be a large number of such components. Therefore, when determining automatic scaling methods, taking into account load forecasting, it is advisable to process historical metrics, as well as adjust model parameters to predict a load of each group of components with the same properties separately.

Actual scientific researches and issues analysis. The topic of using both statistical prediction approaches [2] and artificial intelligence-based approaches for

automating the scaling of computing resources of deployed applications is quite well studied [3]. But these works do not consider the issue of implementing the developed solutions into existing systems, in particular the need to prepare and process historical data, adjust algorithms for existing load patterns, and validate the obtained results.

Uninvestigated parts of general matters defining. A large number of scientific works are devoted to the topic of proactive scaling both by statistical methods and based on artificial intelligence, but the vast majority only evaluate the accuracy of the obtained models and such a characteristic as universality – the ability of the model to work with a large number of various load patterns without manual adjustment – remains secondary [4].

The research objective. The purpose of this work is to investigate the practicability and possibility of applying forecasting-at-scale algorithms for predicting loads in a Kubernetes cluster. Through experiments, test the ability of Prophet and Greykite algorithms to predict typical load patterns of cluster components.

Overview of algorithms. Let us consider two main representatives of forecasting-at-scale algorithms [5], namely Prophet and Greykite. But first, let us define what forecasting-at-scale algorithms are. The name «at-scale» in this context has two meanings. First, it is a simple and powerful tool that does not require the user to have deep knowledge of prediction algorithms. It lets us scale prediction pipeline in terms of reducing time for creating more or less accurate model. Secondly, these algorithms allow solving a large number of various forecasting problems, including reliable practical forecasting of time series.

Prophet [6] is a time series forecasting library that was developed at Facebook. The main goal of the development was to create a simple, transparent and understandable model generation algorithm that would allow to quickly obtain reliable predictions.

This algorithm is based on an additive regression model, which has several components.

$$y(t) = g(t) + s(t) + h(t) + e(t), \quad (1)$$

where $g(t)$ – a trend component, $s(t)$ – a seasonal component, $h(t)$ – anomalies, $e(t)$ – an error function. In addition to the additive regression model, Prophet also uses a Fourier transform.

Among the advantages of this model it has the possibility of working with various time series, the ability to work effectively with large data sets and missing data, as well as flexibility in the setting.

The Greykite Library [7] is a powerful time series forecasting library developed by LinkedIn. Its main task is to provide a flexible, fast, and scalable solution for generating forecasts based on a large amount of data.

Greykite uses the Silverkite model, which, like Prophet, is an additive regression model with several components [8]:

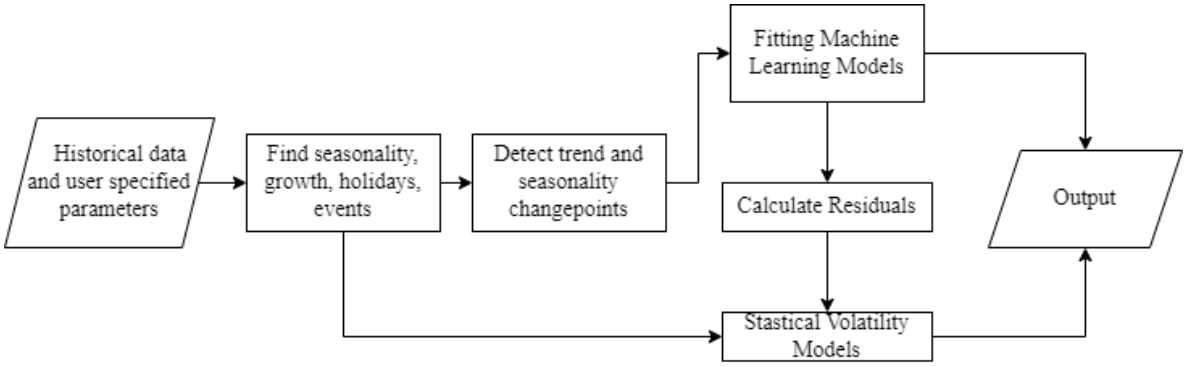

Fig. 1. Greykite's work algorithm

One of the unique features of Greykite is the ability to flexibly configure trending and seasonal components to adapt to different patterns in the data. In addition, Greykite uses different machine learning algorithms to process different components of the model, which makes this solution versatile.

Experiments. To check the accuracy and universality of the models, two typical load patterns were chosen [9]. The first pattern has weekly and daily load fluctuations, the second one has weekly on/off seasonality. Weekly seasonality is chosen because this period of time reflects the repeatability of people’s actions throughout the day, including in the business environment: working hours, time for rest, sleep, and so on. In general, any other seasonality can be chosen, and the goal is to test the performance of the models on complex seasonalities. It is important to clarify that the obtained models should not be adjusted to a specific time series in any way, since it is important to check the universality of the approaches.

To assess the accuracy of time series forecasting models, it is appropriate to use two accuracy metrics – RMSE (root mean square deviation) and MAPE (average absolute error in percentage).

RMSE allows you to compare the deviations of the original values and helps to assess the overall accuracy of the prediction. RMSE is calculated using the following formula:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (\hat{x}_t - x_t)^2}{n}}, \quad (2)$$

where $x(t)$ – an actual value at the moment of time t, $\hat{x}_t$ – prediction at the moment of time t, and n – a number of datapoints in the dataset.

The mean absolute percentage error (MAPE) makes it possible to compare the predictions of different models at different scales or data. MAPE is calculated as follows

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{\hat{x}_t - x_t}{x_t} \right|, \quad (3)$$

where $x(t)$ – an actual value at the moment of time t, $\hat{x}_t$ – prediction at the moment of time t, and n – a number of datapoints in the dataset.

The assessment of the universality of the method is determined on the basis of the assessment of accuracy without additional adjustment of the models. It is important to determine how the model can adapt to various load patterns.

In the first experiment, the selected models are compared on the example of the above-described time series with two periodicities of different lengths – daily and weekly. The data has not been pre-processed. The purpose of this experiment is to investigate the prediction capabilities of the selected models on complex load patterns without any data distortion, and also to investigate the effect of the size of the historical data during training on the prediction accuracy.

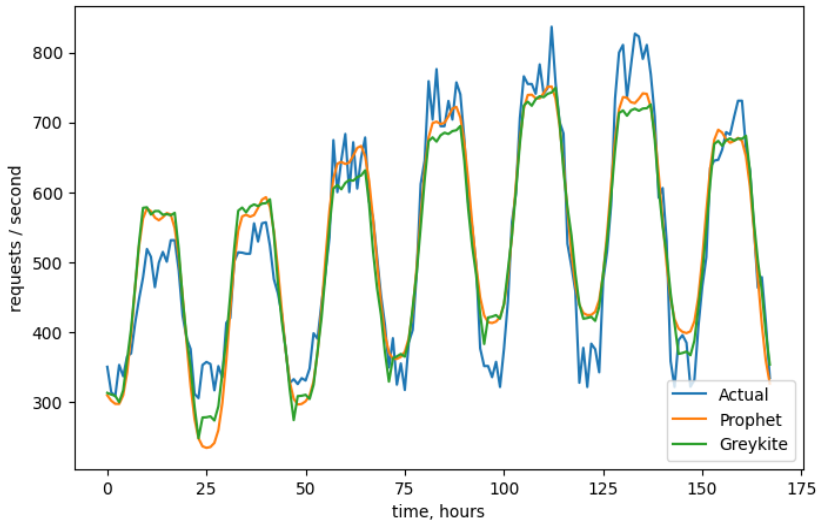

Fig 2. Model predictions based on three-week historical data

In fig. 2 shows the performance results of the selected algorithms after training for three weekly periods. The MAPE indicators are approximately at the same level – 0.085, which indicates high accuracy of the results. It is worth noting that the minimum length of historical data should be at least two target periods for pattern detection. It is appropriate to check the accuracy on the minimum permissible period.

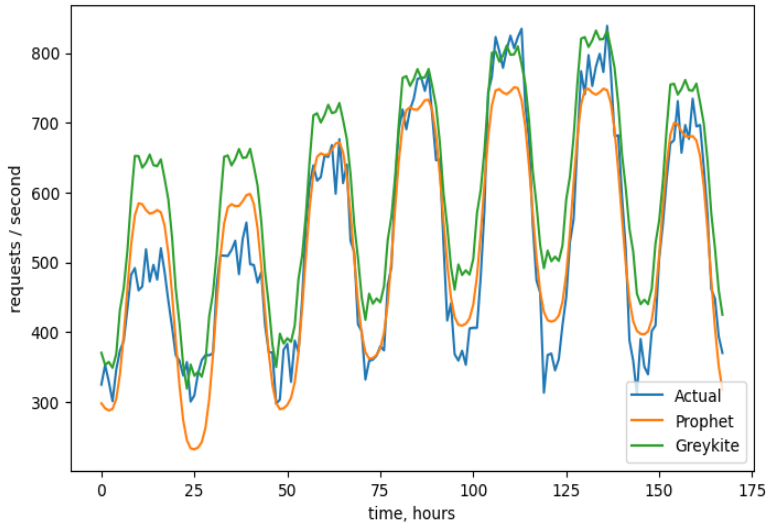

Fig 3. Model predictions based on two-week historical data

In this case, Greykite's accuracy dropped significantly to 0.16, while Prophet remained at the same level of 0.085.

The last experiment is conducted for another pattern, namely weekly on/off, where the load appears quickly and disappears quickly.

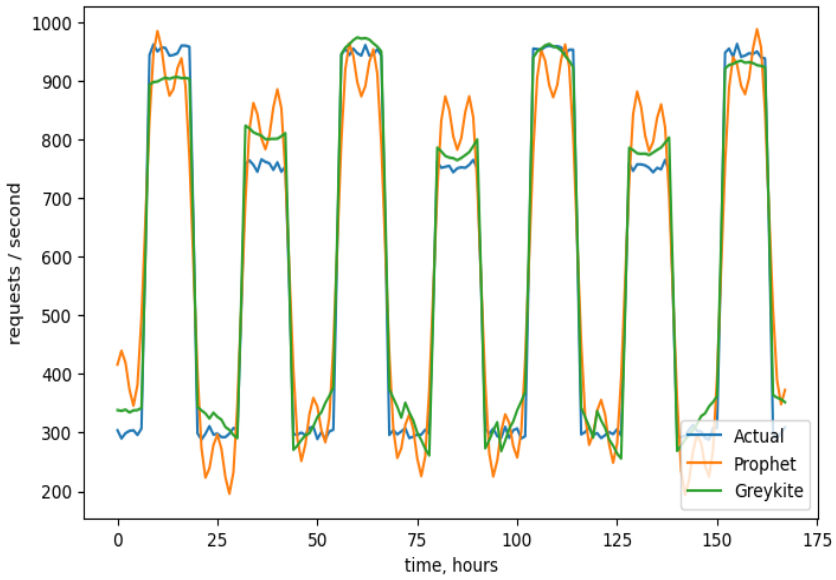

Fig 4. Prediction of on/off pattern patterns

In this experiment, Greykite is extremely accurate and has a MAPE of 0.06. In comparison, MAPE for Prophet is 0.12. All final accuracy indicators are shown in Table 1.

Table 1. **Prophet and Greykite performance results**

Approach	RMSE 3 weeks	MAPE 3 weeks	RMSE 2 weeks	MAPE 2 weeks	RMSE on/off	MAPE on/off
Prophet	49.2857	0.08723	51.1198	0.08929	72.7108	0.12620
Greykite	49.0042	0.08400	88.4236	0.16385	34.1663	0.06612

Conclusions. The experiments conducted in this work prove that the forecasting-at-scale algorithms for predicting time series Prophet and Greykite are appropriate to use when developing solutions for managing the autoscaling of computing resources in a cluster. Both algorithms showed high accuracy and relative versatility. Prophet is more accurate on shorter historical data lengths, and Greykite did much better with the on/off load pattern.

In future works, it is advisable to integrate these algorithms and reactive management approaches [10] into the resource distribution subsystem and conduct similar experiments in real conditions.

References

1. Rolik, A. I., Telenyk, S. F., Yasochka, M. V. (2018). Upravlenye korporatyvnoi ynfrastrukturoi. Kyiv: Naukova Dumka. 576 p.

2. De Livera, M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. In Journal of the American Statistical Association (Vol. 106, No. 496, pp. 1513-1527).

3. Rolik, O., Kolesnik, V., & Halushko, D. (2015). Neural network approach for resource allocation in IT infrastructure Management System. In Proc. Of the Congress on Information Technology, Computational and Experimental Physics 2013 (CITCEP'15) (pp. 176-179). 18–20 December, Cracow, Poland.

4. Lorido-Botrán, T., Miguel-Alonso, J., & Lozano, J. (2014). A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. In Journal of Grid Computing (Vol. 12).

5. Taylor, S. J., & Letham, B. (2018). Forecasting at scale. In The American Statistician (Vol. 72, No. 1, pp. 37-45).

6. Facebook. Prophet: Automatic Forecasting Procedures. Available: https://github.com/facebook/prophet [Accessed: June 21, 2023].

7. LinkedIn. Greykite. Available: https://github.com/linkedin/greykite [Accessed: June 21, 2023].

8. Hosseini, R., Chen, A., Yang, K., Patra, S., Su, Y., Al Orjany, S. E., Tang, S., & Ahammad, P. (2022). Greykite: Deploying Flexible Forecasting at Scale at LinkedIn. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3007-3017).

9. Kim, I. K., Wang, W., Qi, Y., & Humphrey, M. (2016). Empirical Evaluation of Workload Forecasting Techniques for Predictive Cloud Resource Scaling. In 2016 IEEE 9th International Conference on Cloud Computing (CLOUD) (pp. 1-10). San Francisco, CA, USA.

10. Omelchenko, V., & Rolik, O. (2022). Automation of resource management in information systems based on reactive vertical scaling. In "Adaptive Systems of Automatic Control" Interdepartmental scientific and technical collection (No. 2 (41), pp. 65-78).

AUTHORS

Rolik Oleksandr Ivanovych – professor, Doctor of Technology, head of the Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: arolick@gmail.com

Omelchenko Vitalii – PhD student, Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: vitaly.om25@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

Віталій Омельченко, Олександр Ролік

FORECASTING-AT-SCALE АЛГОРИТМИ ДЛЯ ПЕРЕДБАЧЕННЯ НАВАНТАЖЕНЬ В КЛАСТЕРІ

Постановка проблеми. Прі управлінні якістю послуг, які надають хмари або ІТ-інфраструктури корпоративного рівня, важливо підтримувати якість послуг на узгодженому рівні [1]. Здебільшого підтримання якості послуг на узгодженому рівні забезпечується шляхом надання додаткових обчислювальних, комунікаційних та інших ресурсів тім застосункам, які забезпечують відповідний сервіс. Ефективне управління якістю послуг може бети забезпечено лише автоматичним додаванням або зменшенням обсягів ресурсів. Для цього системи управління обчислювальними ресурсами ЦОД хмарних провайдерів або корпоративною ІТ-інфраструктурою містять модулі, які використовують методи та засоби автоматичного масштабування з врахуванням технологій надання сервісів. Зрозуміло, що більш ефективними з точки зору підтримання якості сервісів на узгодженому рівні будуть ті методи автомасштабування, які використовують результати прогнозування навантаження. Сучасні парадигми надання сервісів побудовані на використанні мікросервісів, контейнерів та ін. Популярність використання мікросервісів при наданні послуг потягнула за собою створення різноманітних технологій реалізації мікросервісної архітектури, у тому числі і на основі кластерів. При цьому вирішення задач підтримання якості сервісів на узгодженому рівні доцільно здійснювати шляхом масштабування всіх компонентів кластеру. При цьому необхідно зважати на те, що кожен компонент мікросервісної архітектури має свої унікальні особливості роботи та функціонал, а також на те, що шаблон навантаження є індивідуальним для кожного компоненту, а таких компонентів може бути велика кількість. Тому при визначенні методів автоматичного масштабування з врахування прогнозування навантаження доцільно здійснювати обробку історичних метрик, а також налаштування параметрів моделі для передбачення навантаження кожної групи компонентів з однаковими властивостями окремо.

Аналіз останніх досліджень і публікацій. Тема застосування як статистичних підходів передбачення [2], так і на основі штучного інтелекту для автоматизації масштабування обчислювальних ресурсів кластеру розгорнутих

додатків є досить добре вивченою [3]. Але в цих роботах не розглядається Але в цих роботах не розглядається питання впровадження розроблених рішень в існуючі системи, зокрема необхідність підготовки та обробки історичних даних, налаштування алгоритмів під існуючі шаблони навантаження та валідацію отриманих результатів.

Виділення недосліджених частин загальної проблеми. Темі проактивного масштабування як статистичними методами, так і на основа штучного інтелекту, присвячена велика кількість наукових праць, проте переважна більшість проводить лише оцінку точності отриманих моделей, а така характеристика, як універсальність – здатність моделі працювати з великим числом різноманітних шаблонів навантаження без ручного налаштування залишається другорядною [4].

Мета дослідження. Метою даної роботи є дослідження доцільності та можливості застосування forecasting-at-scale алгоритмів для передбачення навантажень в Kubernetes кластері. Шляхом експериментів перевірити здатність алгоритмів Prophet та Greykite передбачати типові шаблони навантаження компонентів кластеру.

Викладення основного матеріалу. В роботі розглядається архітектура роботи двох застосування forecasting-at-scale алгоритмів – Prophet та Greykite. Описуються умови проведення експериментів та дані для них. Проводяться 3 експеримента з різною довжиною історичних даних та шаблонами навантаження.

Висновки. Проаналізовано доцільність застосування forecasting-at-scale алгоритмів для передбачення робочих навантажень кластеру. Дані алгоритми показали здатність точно передбачати робочі навантаження з різними шаблонами даних і комплексною сезонністю.

Ключові слова: управління ресурсами, проактивне масштабування, Kubernetes, автоматичне масштабування, Prophet, Greykite. Рис.: 4. Табл.: 1. Бібл.: 10.